UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/603,700	06/25/2003	Richard J. Moore	BEA920030012US1	3413

49474 7590 12/27/2007
LAW OFFICES OF MICHAEL DRYJA
1474 N COOPER RD #105-248
GILBERT, AZ 85233

EXAMINER	
STEELMAN, MARY J	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
12/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

MN

Office Action Summary	**Application No.** 10/603,700	**Applicant(s)** MOORE ET AL.	
	Examiner MARY STEELMAN	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *8/30/07, 9/24/07*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-15 and 17-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15 and 17-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

MARY STEELM
PRIMARY EXAM

[illegible]

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *08/30/2007*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to claims, amendments, and remarks received 08/30/2007 and RCE received 09/24/2007. Per Applicant's request, claim 16 has been cancelled. Claims 1, 2, 14, and 18 have been amended. Claims 1-15 and 17-20 are pending. IDS received 08/30/2007 has been considered.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 18-20 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claim 18 recites, " An article of manufacture comprising a machine readable medium..." A machine readable medium is defined in the Specification to include "a modulated carrier signal" at page 5 1st paragraph. Claim may be amended to restrict limitations to statutory embodiments by reciting, "An article of manufacture comprising a recordable medium..."

Response to Arguments

3. Applicant's arguments with respect to claims have been considered but are moot in view of the new grounds of rejection.

Applicant has argued, in substance, the following:

Regarding independent claims 1, 14, and 18, as noted on page 6, Morshed does not anticipate newly amended claim limitations: "the probe program executed and interpreted by an interpreter, the interpreter running on one or more processors, the probe program independent of an architecture of the one or more processors, and the probe program generated from source code written in a high-level language.."

Applicant points out that Morshed does not disclose a probe program that is executed and interpreted by an interpreter (Morshed's interpreter calls the dll, but does not execute the dll.), and that Morshed does not disclose a probe program independent of the architecture of the processors.

Examiner's Response: New art has been provided to reject the amended limitations.

Claim Objections

4. Claim 2 is objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claims, or amend the claims to place the claims in proper dependent form, or rewrite the claims in independent form. "The interpreter" of claim 2 is disclosed in claim 1, line 5.

Claim Rejections - 35 USC § 102

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-4 and 10-13 are rejected under 35 U.S.C. 102(e) as being anticipated by USPN 5,901,315 to Edwards et al.

Per claim 1:

A system comprising:

-one or more processors;

Edwards: FIG. 1, #12, processor

-a base program executable by the one or more processors and having one or more breakpoints;

Edwards: FIG. 4, breakpoint is hit & related thread events. Col. 4: 8-10, manipulate the target application...breakpoint events

-a probe program associated with each breakpoint, the probe program executed and interpreted by an interpreter, the interpreter running on one or more processors, the probe program independent of an architecture of the one or more processors, and the probe program generated from source code written a high-level language, that is executable by the one or more processors via an interpreter, independent of an architecture of the one or more processors, and generated from source code written in a high-level language, Edwards: Col. 4: 47, the component that consists of the Java code is the javaprobe process 36. Col. 4: 51-54, the javaprobe process 36 uses the Java debug API to control and manipulate the Java portions of the application under debug. Col. 5: 3-8, "the javaprobe process preferably comprises Java code (independent of an architecture / source code written in a high level language). The javaprobe process may use the RemoteDebugger Java class (part of the JVM) (an interpreter) to manipulate the target application."

-the probe program associated with each breakpoint being executed when the breakpoint is reached during execution of the base program.

Edwards: Col. 6: 24-25, "user has set a Java breakpoint and resumed the system..." FIG. 4 & related text at col. 6: 36-41, the breakpoint is hit. This causes the breakpointEvent callback in the Java thread to be invoked…At state 42, the javaprobe (probe program associated with each breakpoint) sends a breakpoint notification to jdaemon and returns from the callback…"

Per claim 2:

-further comprising the interpreter.

Edwards: Col. 5: 1-8, JVM (interpreter)

Per claim 3:

-the base program has a first address space and the probe program associated with each breakpoint has a second address space different from the first address space.

Edwards: See FIG. 3 (address space divided by dashed line) and related text at col. 4: 46. Col. 5: 39, "addressing scheme to distinguish between Java and 'C'..."

Per claim 4:

-one or more probe expressions that address objects of the base program in the first address space and that are used by objects of the probe program in the second address space to communicate with the objects of the base program in the first address space.

Edwards: See FIGs. 5A-5B, Host Machine (first address space), Target Machine (second address space), javaprobe in Target Machine, null terminated string (probe expression) passed to JAVA EE (expression evaluator), communication passes string via TCP/IP, socket

Per claim 10:

-probe program associated with each breakpoint is able to pass user messages by manipulating a state of one of the probe program and base program.

Edwards disclosed (col. 7: 18-50) the probe program associated with each breakpoint passes messages. "The jdaemon and javaprobe processes know whether a particular debug event originates in the native method or JAVA code..." Col. 7: 53-65, "the debug engine (DE) interacts with the jdaemon process throughout the debug session. When the DE needs to evaluate an expression, for example, the context of the expression determines which language specific evaluator to se. The DE communicates through a common interface to each expression evaluator (EE). This common interface defines requests (user messages) the DE can make…"

Col. 8: 7-15, "a stub JAVA EE on the host machine which bundles any DE request (user messages) into a character string and makes a service request...In order to handle a particular request (manipulate state of probe program and base program), the JAVA EE may need to communicate with the JVM via the RemoteDebugger class and its associated objects (stack, variable information, etc.)."

Col. 8:19-21, service requests an EE can make (read/write memory, read/write registers, get debug information, etc.)" (manipulate state of probe program and base program).

Per claim 11:

-probe program associated with each breakpoint is able to pass user messages by manipulating a stack of one of the probe program and base program.

Edwards: Col. 8: 13-14, "communicate with the JVM via the RemoteDebugger class and its associated objects (stack, variable information, etc.)."

Per claim 12:

-the base program is written in a high-level language different than the high-level language in which the probe program associated with each breakpoint is written.

Edwards: Col. 4: 39-41, C/C++ & JAVA

Per claim 13:

-the base program is written in a high-level language that is identical to the high-level language in which the probe program associated with each breakpoint is written.

Edwards: Col. 4: 41, target JAVA application Col. 4: javaprobe process 36, JAVA code Col. 6: 25, JAVA breakpoint

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 5-9 & 14-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over USPN 5,901,315 to Edwards et al., in view of USPN 6,738,967 B1 to Radigan.

Per claim 5:

-a high-level language compiler to compile the probe program from source code written in the high-level language to object code.

Edwards: Col. 3: 55-56, "debug a target application that is written in JAVA...: Col. 5: 3-8, "the javaprobe process preferably comprises Java code (independent of an architecture / source code written in a high level language)."

Edwards failed to explicitly disclose "compile the probe program from source code written in the high-level language to object code." However, it is noted that an executable JAVA code is compiled into bytecode for interpretation via a JVM.

More explicitly, Radigan disclosed (col. 6: 46-56), "optimizing VM to translate and optimize the IL program...Such just-in-time translators ('jitters') can be made fast enough that a user does no perceive a significant delay between translator invocation and object-program execution." Col. 8: 54-58, "A virtual machine in each user system includes a code translator component 251 that receives the IL program 222 of package 225 and converts it through JIT compiling ...or otherwise translating it to object language (OL) program 254a-n in the form of individual instruction in the object code or language of processor 255a-n..." Radigan disclosed (col. 4: 5-17) a development system...compiles ...source code...into an intermediate language (IL) or tokenized form of low level instructions for an abstract machine (VM) and provides information in edges of the graph, concerning complex relationships...The IL program itself is independent of any particular processor or architecture and contains processor independent constructs, in the form of annotations...Processor specific annotations are also provided, which identify possible, legal, profitable, machine specific, optimizations...

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Edwards, using the teachings of Radigan, because Radigan recognized that an optimized, compiled JAVA program (col. 3: 4) speeds performance. Edwards disclosed (col. 1: 47-52) debugging an application written in JAVA and extended using native method dynamic load library functions. Jitting such an application, would obviously improve speed in the development process.

Per claim 6:

-the high-level language compiler is able to seamlessly intermix variables of the base program in the first address space and variables of the probe program in the second address space.

Edwards disclosed intermixing (col. 3: 56-61) in "a target application that is written in JAVA but that optionally comprises one or more native method dynamic load libraries or dll's... simultaneously debugging JAVA and C/C++ code."

More explicitly, Radigan disclosed (col. 8: 19-22), "Additional sets of components 220 can be developed for additional source languages, such that programs of different source languages can all be compiled down to a common IL program representation." Radigan disclosed annotated intermediate language, parsed into a tree (col. 7: 21), converted to a common intermediate language and object code (col. 8:14).

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Edwards, using the teachings of Radigan, because Radigan recognized that an optimized, compiled JAVA program (col. 3: 4) speeds performance. Edwards disclosed (col. 1: 47-52) debugging an application written in JAVA and extended using native method dynamic load library functions, a mixed source code application. Jitting such an application would obviously improve speed in the development process. Radigan disclosed a combination of source code, compiled into a common intermediate language. Special annotations on the parse tree are used to make the object code (col. 8: 13-14) for a particular processor.

Per claim 7:

-the probe program associated with each breakpoint is generated from an abstract syntax tree (AST) used to switch between the first and the second address spaces.

Edwards disclosed (col. 5: 30-35), "the jdaemon process 34 manipulates the C/C++ threads of the target application's JVM to allow icatjvo 30 to step into C/C++ methods (switch between the first and second address spaces) Col. 5: 38-41, "the jdaemon process launches the JVM under the system debug API. By using DosDebug to control a JVM and an addressing scheme to distinguish between JAVA and "C" (or C++) native events, one can debug a JAVA application and its native methods with the same debugger at the same time."

Edwards failed to disclose "the probe program associated with each breakpoint is generated from an abstract syntax tree (AST)..."

However, Radigan disclosed an (col. 5: 40-48) "annotating compiler program 140...parses source program 130 to an annotated intermediate language program 150. intermediate language program 150 comprises an intermediate form or parse tree (abstract syntax tree) express in any of a number of existing or purpose built intermediate languages and annotations. The intermediate language representation contains processor independent constructs / graph edges …Processor dependent annotations identify which of these optimizations are most suited for a particular processor architecture…" Radigan disclosed a virtual machine (col. 6: 20-23) that translates an intermediate language program 150 into an object come program. Col. 7: 21-22, "parse tree whose nodes define variables or perform operations, or both."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Edwards, using the teachings of Radigan, because Radigan recognized that an optimized, compiled JAVA program (col. 3: 4) speeds performance. Edwards disclosed (col. 1: 47-52) debugging an application written in JAVA and extended using native method dynamic load library functions, a mixed source code application. Jitting such an application would obviously improve speed in the development process. Radigan disclosed a combination of source code, compiled into a common intermediate language. Special annotations on the parse tree are used to make the object code (col. 8: 13-14) for a particular processor.

Per claim 8:

Edwards failed to disclose:

-the AST has a plurality of nodes.

However, Radigan disclosed an (col. 5: 40-48) "annotating compiler program 140…parses source program 130 to an annotated intermediate language program 150. intermediate language program 150 comprises an intermediate form or parse tree (abstract syntax tree) express in any of a number of existing or purpose built intermediate languages and annotations. The intermediate language representation contains processor independent constructs / graph edges ...Processor dependent annotations identify which of these optimizations are most suited for a particular processor architecture..." Radigan disclosed a virtual machine (col. 6: 20-23) that translates an intermediate language program 150 into an object come program. Col. 7: 21-22, "parse tree whose nodes (plurality of nodes) define variables or perform operations, or both."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Edwards, using the teachings of Radigan, because Radigan recognized that an optimized, compiled JAVA program (col. 3: 4) speeds performance. Edwards disclosed (col. 1: 47-52) debugging an application written in JAVA and extended using native method dynamic load library functions, a mixed source code application. Jitting such an application would obviously improve speed in the development process. Radigan disclosed a combination of source code, compiled into a common intermediate language. Special annotations on the parse tree are used to make the object code (col. 8: 13-14) for a particular processor.

Per claim 9:

Edwards failed to disclose:

-the AST is able to be serialized into an interim format and deserialized from the interim format to reconstruct the AST.

However, Radigan disclosed (col. 12: 26-29), "the distribution of both the intermediate program 222 and the processor dependent annotations 224. Although it is foreseen that these two items will most often be distributed together as a single package (serialized)... Col. 3: 35-43, "The executable program is then distributed (serialized) to multiple different systems. Every system that can execute this type of program uses a light weight translator (deserialized) that produce machine specific code optimized by use of the embedded graph structure for that system. Sets of machine specific annotations are also provided to specifically identify use of the graph to optimize the program."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Edwards, using the teachings of Radigan, because Radigan recognized that an optimized, compiled JAVA program (col. 3: 4) speeds performance. Edwards disclosed (col. 1: 47-52) debugging an application written in JAVA and extended using native method dynamic load library functions, a mixed source code application. Jitting such an application would obviously improve speed in the development process. Radigan disclosed a combination of source code, compiled into a common intermediate language. Special annotations on the parse tree are used to make the object code (col. 8: 13-14) for a particular processor.

Per claim 14:

A method for constructing and using a probe program associated with a breakpoint of a base program comprising:

-constructing an abstract syntax tree (AST) having a plurality of nodes;

-representing objects of the base program with at least some of the nodes of the AST;

-representing objects of the probe program with other of the nodes of the AST;

-switching between a first address space of the objects of the base program and a second address space of the objects of the probe program by traversing the AST.

-serializing the AST into an interim format and storing the AST as serialized into the interim format,

-such that the probe program is executable via an interpreter, and is independent of an architecture of one or more processors executing the base program.

See rejection of limitations as addressed in claims 1 & 9 above.

Per claim 15:

-comprising: deserializing the AST from the interim format to reconstruct the AST.

See rejection of limitations as addressed in claim 9 above.

Per claim 17:

-manipulating a stack of the base program to pass user messages.

See rejection of limitations as addressed in claim 11 above.

Per claim 18:

An article of manufacture comprising:

-a machine-readable medium;

-means in the medium for probing a base program at a breakpoint thereof in a processor architecture-independent manner, an abstract syntax tree (AST) constructed by the probe program, serialized into an interim format, and stored.

See rejection of limitations addressed in claims 1 & 9 above.

Per claim 19:

-the means is written in a high-level language, and employs the AST.

See rejection of limitations addressed in claim 1 above.

Per claim 20:

-the medium is a recordable data storage medium.

Edwards: Col. 8: 63 – col. 9:13.

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached at (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned: 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

12/13/2007

MARY STEELMAN
PRIMARY EXAMINER

[illegible]